2006-06-14

Practical Stabilization Through Real-Time Optimal Control, ACC (2006; Minneapolis, Minnesota)

Ross, I. Michael

IEEE

Practical Stabilization Through Real-Time Optimal Control

I. Michael Ross[1], Qi Gong[1], Fariba Fahroo[2] and Wei Kang[2]

Abstract— **Infinite-horizon, nonlinear, optimal, feedback control is one of the fundamental problems in control theory. In this paper we propose a solution for this problem based on recent progress in real-time optimal control. The basic idea is to perform feedback implementations through a domain transformation technique and a Radau based pseudospectral method. Two algorithms are considered: free sampling frequency and fixed sampling frequency. For both algorithms, a theoretical analysis for the stability of the closed-loop system is provided. Numerical simulations with random initial conditions demonstrate the techniques for a flexible robot arm and a benchmark inverted pendulum problem.**

I. Introduction

We consider the problem of seeking feedback solutions to the following optimal control problem,

$$(\mathsf{B}) \begin{cases} \text{Minimize} & J[x(\cdot),u(\cdot)] = \displaystyle\int_{t_0}^{\infty} F(x(t),u(t))\ dt \\ \text{Subject to} & \dot{x}(t) = f(x(t),u(t)) \\ & x(t_0) = x_0 \\ & h(x(t),u(t)) \leq 0 \end{cases}$$

where $t \mapsto (x,u) \in \mathbb{R}^{N_x} \times \mathbb{R}^{N_u}$ is the state-control function pair, and $F : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \to \mathbb{R}$, $f : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \to \mathbb{R}^{N_x}$ and $h : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \to \mathbb{R}^{N_h}$, are continuously differentiable with respect to their arguments. Except for special cases, such as linear-quadratic problems, no analytical solutions have been found for Problem B. The difficulties in solving this problem stem from the difficulties in solving the associated Hamilton-Jacobi-Bellman (HJB) equation. An alternative to solving the HJB equation is to seek real-time open loop solutions to Problem B. This simple and intuitive concept circumvents the difficulties associated with the HJB equation but relies heavily on modern computational capabilities to generate real-time solutions. In an effort to circumvent the computational burden, and other issues arising in online solutions to Problem B, receding horizon control strategies have been proposed and constitute an active area in control theory [10], [11]. These methods basically approximate the infinite horizon problem by a sequence of sub-optimal receding finite-horizon problems. The main issue in such methods is the stability of the closed-loop system. Among many, one of the reasons the stability problem takes center stage in such methods is due to the use of a finite-horizon predictive control over an infinite horizon. In addition, computational delay, sampling time, control quantization and a host of other issues are prominent in receding horizon control theory [10], [11]. In this paper, we propose a far simpler approach to feedback control. The idea is still based on online optimization, but instead of receding the horizon, we move the origin while keeping the horizon fixed at infinity. This concept is facilitated by a time-domain transformation technique [1], [3] that preserves the notion of the infinite horizon. The accuracy issue is addressed by using pseudospectral (PS) methods that are increasingly popular, particularly in aerospace applications; see for example, [2], [5], [7], [13], [16], [18], [19] and the references therein. Since PS methods are capable of generating optimal solutions in fractions of a second [14], [15], [19], feedback through real-time optimal control is possible by way of non-analytical output-to-input maps.

Based on these results, we explore two algorithms: one is a free-sampling frequency approach [17] that maximally exploits the computational power, and another based on a fixed sampling frequency [4] that maximally exploits prediction with online optimization to reduce the effects of computational delay. In the absence of computational error, stability is guaranteed by Bellman's Principle of Optimality. Preliminary theoretical analysis ensures practical stability in the presence of computational error and disturbances.

II. Pseudospectral Methods for Infinite-Horizon Optimal Control

In this section, we summarize the computational methods from [3] for solving Problem B. The key idea is to use domain transformation to map the semi-infinite domain to the half-open, finite interval, $[-1,1)$, and then use the appropriate PS discretization scheme. In the following, we focus on the rational mapping and the Legendre PS method. Similar ideas apply to other domain transformations and orthogonal polynomials.

For $t \in [0,\infty)$ and $\tau \in [-1,1)$ let

$$t = \frac{1+\tau}{1-\tau} \ \Leftrightarrow\ \tau = \frac{t-1}{t+1} \tag{1}$$

Using (1) and its derivative,

$$\frac{dt}{d\tau} = \frac{2}{(1-\tau)^2} := r(\tau)$$

we reformulate Problem B on the finite interval $[-1,1)$. For the purpose of simplicity, we abuse notation in not distinguishing between $t(\tau)$ and τ for functional dependencies and state the transformed problem as: determine the state-control function pair $[-1,1] \ni \tau \mapsto \{x \in \mathbb{R}^{N_x}, u \in \mathbb{R}^{N_u}\}$ that minimizes the cost functional,

$$J[x(\cdot),u(\cdot)] \ = \ \int_{-1}^{1} F(x(\tau),u(\tau))r(\tau)\ d\tau$$

This research was supported in part by NPS and the Secretary of the Air Force.

[1]Department of Mechanical and Astronautical Engineering, Naval Postgraduate School, Monterey, CA, 93943, imross,qgong@nps.edu

[2]Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA, 93943, ffahroo,wkang@nps.edu

subject to the dynamics,

$$\frac{dx}{d\tau} \;=\; r(\tau) f(x(\tau), u(\tau))$$

initial conditions $x(-1) = x_0$ and path constraints $h(x(\tau), u(\tau)) \leq 0$. It should be emphasized that in this formulation, all functional evaluations at $\tau = 1$ which corresponds to the value of the original function at $t = \infty$ is taken in the sense of a limit [3].

As presented in [3], in the Radau-based Legendre PS method, we approximate the trajectory by N-th order Lagrange interpolating polynomials over Legendre-Gauss-Radau (LGR) nodes, i.e.

$$x(\tau) \;\approx\; x^N(\tau) \;:=\; \textstyle\sum_{k=0}^{N} x^N(\tau_k)\phi_k(\tau),$$

where $\phi_k(\tau)$ are the Lagrange interpolating polynomials satisfying $\phi_k(\tau_j) = 1$, if $k = j$ and $\phi_k(\tau_j) = 0$, if $k \neq j$. The LGR nodes, τ_k, are defined by the initial point, $\tau_0 = -1$, and as zeros of $L_N + L_{N+1}$ where L_N is the Legendre polynomial of degree N. For these points which are distributed over $[-1, 1)$, evaluation at the right-hand point (which for the mapped domain corresponds to ∞) is at $\tau_N = 1 - \epsilon$, where the size of ϵ depends inversely on N; that is, $\epsilon \to 0$ as $N \to \infty$. It is worth mentioning that the distribution of the LGR nodes on $[-1, 1)$ is much denser near the -1 end than near the $+1$ end. This feature favors closed-loop control since only the control signal close to the initial node is implemented. This point will be clearer later.

The derivative of the i-th state $x_i(\tau)$ at the LGR node τ_k is approximated by

$$\dot{x}_i(\tau_k) \;\approx\; \dot{x}_i^N(\tau_k) = \textstyle\sum_{j=0}^{N} D_{kj} x_i^N(\tau_j),\; i = 1, 2, \ldots, N_x$$

where the $(N + 1) \times (N + 1)$ differentiation matrix D is defined by $D_{kj} = \dot{\phi}_j(\tau_k)$. Let $\bar{x}_k = x^N(\tau_k)$, $k = 0, 1, \ldots, N$. The continuous differential equation is approximated by the following nonlinear algebraic equations

$$\sum_{i=0}^{N} \bar{x}_i D_{ki} \;=\; r(\tau_k) f(\bar{x}_k, \bar{u}_k), \qquad k = 0, 1, \ldots, N \tag{2}$$

The initial conditions and constraints are approximated in a similar fashion

$$\bar{x}_0 \;=\; x_0 \tag{3}$$

$$h(\bar{x}_k, \bar{u}_k) \;\leq\; 0, \quad k = 0, 1, \ldots, N \tag{4}$$

Finally, the cost functional $J[x(\cdot), u(\cdot)]$ is approximated by the Gauss-Radau integration rule,

$$J[x(\cdot), u(\cdot)] \;\approx\; \bar{J}^N(\bar{X}, \bar{U}) \;=\; \textstyle\sum_{k=0}^{N} F(\bar{x}_k, \bar{u}_k) r(\tau_k) w_k$$

where w_k are the LGR weights given by

$$w_0 \;=\; \frac{2}{(N+1)^2}, \qquad w_j \;=\; \frac{1}{(N+1)^2} \frac{1 - \tau_j}{[L_N(\tau_j)]^2}$$

and $\bar{X} = [\bar{x}_0, \ldots, \bar{x}_N]$, $\bar{U} = [\bar{u}_0, \ldots, \bar{u}_N]$. Thus, the optimal control Problem B is approximated to a nonlinear programming problem with $\bar{J}^N$ as the objective function and (2), (3) and (4) as constraints; this is summarized as:

$$(B^N) \begin{cases} \text{Minimize} & \bar{J}^N(\bar{X}, \bar{U}) \;= \displaystyle\sum_{k=0}^{N} F(\bar{x}_k, \bar{u}_k) r(\tau_k) w_k \\ \text{Subject to} & \displaystyle\sum_{i=0}^{N} \bar{x}_i D_{ki} - r(\tau_k) f(\bar{x}_k, \bar{u}_k) \;=\; 0 \\ & \qquad k = 0, 1, \ldots, N \\ & h(\bar{x}_k, \bar{u}_k) \;\leq\; 0, \\ & \bar{x}_0 \;=\; x_0 \end{cases}$$

This problem can now be solved by a robust, spectral algorithm [6]. These ideas are implemented in an α-version of the software package, DIDO [12]. All of the computations reported in this paper were based on this version of DIDO, programmed in MATLAB on a Pentium 4, 2.4GHz PC with 256MB of RAM.

III. Feedback Design Based on PS Methods

A closed-loop controller is constructed as follows: At every sampling instant, a measurement is taken. Based on this measurement, an open loop optimal control is computed using the PS method. This control is applied to the system over some period. The procedure is repeated over the next measurement. Note that this is not a sample-and-hold technique. In accounting for the computational delay, two closed-loop control designs are proposed.

A. Free Sampling Frequency

In this section we propose a feedback strategy when the sampling frequency of the system is not fixed, i.e, the measurement can be taken at any time instant. The basic idea is to apply the computed open loop optimal controller as soon as it is available.

Denote t_i as the current time and $x(t_i)$ as the current state. Based on measurement $x(t_i)$, an optimal open-loop control trajectory, $u_i^*(x(t_i), t)$, is calculated. Suppose that after time $t_i + \Delta t_i$, the controller $u_i^*(x(t_i), t)$ is available. Therefore, we can only apply this control to the system after $t_i + \Delta t_i$. It implies that the next sampling time is chosen as $t_{i+1} = t_i + \Delta t_i$; and correspondingly, we denote Δt_{i+1} as the computational time to get $u_{i+1}^*(x(t_i + \Delta t_i), t)$. With this notation, the closed-loop system can be written down as

$$\begin{aligned} \dot{x} \;&=\; f(x, u(t)) \\ u(t) \;&=\; u_{i-1}^*(x(t_{i-1}), t), \qquad \forall t \in [t_i, t_{i+1}] \\ t_{i+1} \;&=\; t_i + \Delta t_i \end{aligned}$$

for all $i = 1, 2, \ldots$. Initially, the control is set to

$$u(t) \;=\; u_0^*(x_0, t),\; \forall t \in [t_0, t_1],$$

where t_1 is a fixed constant. The closed-loop controller design is summarized in the following algorithm.

Algorithm I (Free Sampling Frequency)

Initialization:

1. Choose discretization parameter N and the first sampling time t_1.
2. Collect the initial state $x_0 = x(t_0)$.
3. Use the PS method to calculate the optimal controller $u^*(x_0, t)$ and apply it to the system till $t = t_1$. Let $i = 1$.

Main Algorithm:

1. Collect the new measurement $x(t_i)$.
2. Use the PS method to calculate the optimal controller $u^*(x(t_i), t)$ and set Δt_i equal to the computation time.
3. Set $t_{i+1} = t_i + \Delta t_i$. During the time period $[t_i, t_{i+1}]$, apply the controller $u^*(x(t_{i-1}), t)$.
4. Set i = i+1 and go to step 1.

In this feedback architecture, the sampling period is not a fixed number. It depends on the computational time at each iteration. In this way, the computational power is maximally explored; and the computational delay is minimized. In the ideal situation where there is no computational error, no model mismatch and no disturbance/noise, by the Principle of Optimality the closed-loop control and the trajectory should exactly follow the open-loop ones. Therefore, the stability of the controlled system can be guaranteed as long as the open loop optimal control can stabilize the system. Note that in the ideal case, we do not need to assume the computational time to be zero. Indeed, the computational delay Δt_i can be arbitrarily large.

Example 1: Consider the one-link robot arm with a flexible joint [8], [14]. The system is modelled by

$$\begin{aligned} \dot{x}_i &= x_{i+1}, \qquad i = 1, 2, 3 \\ \dot{x}_4 &= (\beta \cos x_1 - 2)x_3 - \beta(x_2^2 - 1)\sin x_1 + u \end{aligned}$$

where $\beta = -0.049$. The problem is to minimize

$$J[x(\cdot), u(\cdot)] = \int_0^\infty \sum_{i=1}^4 x_i^2(t) + u^2(t)\ dt$$

subject to the control constraint $|u| \leq 1$. The closed-loop performance of Algorithm I with a discretization parameter $N = 24$ is shown in Figure 1. The initial condition is $x(0) = [1, 0, -1, 0]^T$. An analytic solution to this problem is

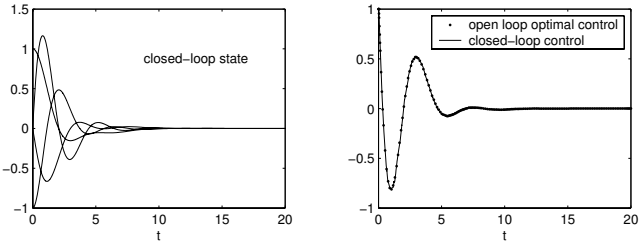

Fig. 1. The closed-loop control and state with 24 nodes. The dotted line in the right figure is the open loop optimal control generated using 100 nodes.

unavailable; hence, a discrete optimal solution with a large number of nodes (N=100) is shown in Figure 1 for reference. Clearly, the closed-loop controller follows the open loop optimal control very well although a substantially smaller number of nodes is used. The average running time for each iteration is 0.09 second.

B. Stability Analysis

In this section, we present some preliminary analysis on the stability of the proposed controller under computational errors.

Assumption 1: The nonlinear vector field $f(x, u)$ satisfies $f(0, 0) = 0$ and is Lipschitz continuous, i.e., there are constants $L_1 > 0$, $L_2 > 0$, such that

$$\|f(x, u) - f(y, v)\| \leq L_1\|x - y\| + L_2\|u - v\|$$

Assumption 2: For any initial condition $x(t_0) = x_0$, Problem B has an optimal solution $u^*(x_0, t)$. Moreover, for the nonlinear system

$$\dot{x} = f(x, u^*(x_0, t)),$$

there is a Lyapunov function $V(x) \in C^1$, satisfying

$$c_1\|x\|^2 \leq V(x) \leq c_2\|x\|^2 \tag{5}$$

$$\frac{\partial V}{\partial x} f(x, u^*(x_0, t)) \leq -c_3\|x\|^2 \tag{6}$$

$$\left\| \frac{\partial V}{\partial x} \right\| \leq c_4\|x\| \tag{7}$$

where c_i, $i = 1, \ldots, 4$, are all positive constants.

Remark 1: The Lyapunov function is not used in the control algorithm. We only need the existence of $V(t, x)$ for the proof of stability. Assumption 2 basically requires the nonlinear system to be uniformly exponentially stable under the open loop optimal control. Furthermore, the Lyapunov function is independent of the initial condition. Indeed, from (5)-(6)-(7) and the Comparison Lemma [9], it is easy to show

$$V(x(t)) \leq e^{-\frac{c_3}{c_2}(t - t_0)} \cdot V(x(t_0)) \tag{8}$$

On the other hand, if the optimal control can be written down in a feedback form, $u^*(\cdot) = u^*(x)$ with $u^*(0) = 0$, and the equilibrium $x = 0$ of the closed-loop system is exponentially stable, by the Lyapunov Converse Theorem, there must exist a Lyapunov function $V(x)$ satisfying (5)-(6)-(7) [9].

Assumption 3: Given any two initial conditions $x(t_0) = z^1$ and $x(t_0) = z^2$, let us denote $u^*(z^1, t)$ and $u^*(z^2, t)$ as the optimal controls with the corresponding initial conditions. It is assumed that

$$\|u^*(z^1, t) - u^*(z^2, t)\| \leq L_3\|z^1 - z^2\|,\ \forall t \in [t_0, \infty]$$

where $L_3 > 0$ is a constant.

Remark 2: This assumption requires that the difference in the optimal controls be linearly bounded by the difference in the initial conditions. If the optimal control can be written down in a feedback form, i.e., $u^*(t) = u^*(x(t))$ and A1 is satisfied, then Assumption 3 is equivalent to $u^*(x)$ being Lipschitz continuous in x.

Let $x(t_i)$ be the state at the sampling time t_i and $u_i^*(t)$, $t \in [t_i, t_{i+1}]$, be the optimal solution to Problem B with $x(t_i)$ as the initial condition. Let $u_i(t)$, $t \in [t_i, t_{i+1}]$, be the control applied to the system by way of Algorithm I. Assuming no computational error, $u_i(t) = u_{i-1}^*(t)$. Further, by the Principle of Optimality, $u_i^*(t) = u_{i-1}^*(t)$. However due to computational errors and disturbances, we have $u_i(t) \neq u_{i-1}^*(t)$. In the following, we assume the computational error to be uniformly bounded.

Assumption 4: There is a positive constant ϵ such that $\forall\ t \in [t_i, t_{i+1}]$

$$\|u_i(t) - u_{i-1}^*(t)\| \leq \epsilon$$

for all $i \geq 0$, and $u_{-1}^*(t)$ is defined as $u_{-1}^*(t) = u_0^*(t)$.

Figure 2 illustrates the concept. Note that the real control applied to the system may be discontinuous. We assume the

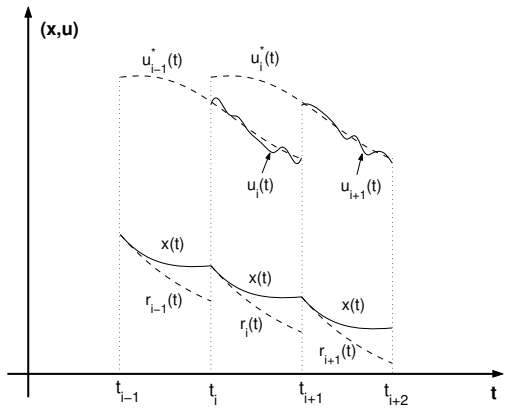

Fig. 2. Control, state and their reference trajectories. $r_i(t)$ is the optimal state trajectory starting from $x(t_i)$ and driven by the optimal control $u_i^*(t)$.

state is absolutely continuous and the control to be piecewise continuous.

Let Δt_i denote the sampling period in step i and Δt_{min}, Δt_{max} represent the minimum and maximum sampling periods.

Proposition 1: Let the controller be designed according to Algorithm I. Under Assumption 1—4, there is a fixed constant ΔT, such that, if $\Delta t_{max} \leq \Delta T$, then for any initial condition $x(t_0)$, the state of the closed-loop system is bounded for all $t \in [t_0, \infty)$. Moreover, there is a positive constant M depending on the system parameters and sampling period, such that

$$\lim_{t\to\infty} \|x(t)\| \quad \leq \quad \epsilon \cdot M$$

Proof: Consider two trajectories $x(t)$ and $r_i(t)$, where $x(t)$ satisfies

$$\begin{aligned} \dot{x}(t) &= f(x(t), u_i(t)), \qquad t \in [t_i, t_{i+1}] \\ x(t_i) &= x(t_i) \end{aligned}$$

and $r_i(t)$ satisfies

$$\begin{aligned} \dot{r}_i(t) &= f(r_i(t), u_i^*(t)), \qquad t \in [t_i, t_{i+1}] \\ r_i(t_i) &= x(t_i) \end{aligned}$$

In other words, $x(t)$ is the system trajectory under the real control $u_i(t)$, and $r_i(t)$ is the optimal trajectory starting from the same initial condition $x(t_i)$ and driven by the optimal control $u_i^*(t)$. The reference trajectory is $r_i(t)$ over each sampling interval $[t_i, t_{i+1}]$.

By the Principle of Optimality, $u_{i-1}^*(t)$, $t \geq t_i$, is the optimal control with the initial condition $x(t_i) = r_{i-1}(t_i)$. Therefore, from A3 and A4, $\forall\ t \in [t_i, t_{i+1}]$

$$\begin{aligned} \|u_i(t) - u_i^*(t)\| &\leq \|u_i(t) - u_{i-1}^*(t)\| + \|u_{i-1}^*(t) - u_i^*(t)\| \\ &\leq \epsilon + L_3 \|x(t_i) - r_{i-1}(t_i)\| \end{aligned}$$

By Assumption 1, it is easy to show

$$\begin{aligned} \|x(t) - r_i(t)\| \ \leq\ & L_2 \Delta t_i [\epsilon + L_3 \|x(t_i) - r_{i-1}(t_i)\|] \\ & + L_1 \int_{t_i}^{t} \|x(\tau) - r_i(\tau)\| d\tau,\ \forall t \in [t_i, t_{i+1}]. \end{aligned} \tag{9}$$

By Gronwall's inequality [9], (9) implies that, $\forall t \in [t_i, t_{i+1}]$

$$\|x(t) - r_i(t)\| \leq L_2 \Delta t_i [\epsilon + L_3 \|x(t_i) - r_{i-1}(t_i)\|] e^{L_1 \Delta t_i}$$

Therefore,

$$\begin{aligned} \|x(t_{i+1}) - r_i(t_{i+1})\| \leq\ & \epsilon L_2 \Delta t_i e^{L_1 \Delta t_i} + \\ & L_2 L_3 \Delta t_i e^{L_1 \Delta t_i} \|x(t_i) - r_{i-1}(t_i)\| \end{aligned}$$

Choose ΔT such that $2L_2 L_3 \Delta T e^{L_1 \Delta T} = 1$. Then, for all $\Delta t_{max} \leq \Delta T$,

$$\begin{aligned} \|x(t_{i+1}) - r_i(t_{i+1})\| \quad \leq \quad & \epsilon L_2 \Delta t_{max} e^{L_1 \Delta t_{max}} + \\ & \frac{1}{2} \|x(t_i) - r_{i-1}(t_i)\| \end{aligned} \tag{10}$$

The formula above is true for all i. Recursively applying (10) leads to

$$\begin{aligned} \|x(t_{i+1}) - r_i(t_{i+1})\| \quad \leq \quad & \epsilon L_2 \Delta t_{max} e^{L_1 \Delta t_{max}} \sum_{k=0}^{i-1} \frac{1}{2^k} \\ & + \frac{1}{2^i} \|x(t_1) - r_0(t_1)\| \end{aligned}$$

By the initialization step of Algorithm I,

$$\|x(t_1) - r_0(t_1)\| \quad \leq \quad \epsilon L_2 \Delta t_{max} e^{L_1 \Delta t_{max}}$$

Hence

$$\|x(t_{i+1}) - r_i(t_{i+1})\| \quad \leq \quad 2\epsilon L_2 \Delta t_{max} e^{L_1 \Delta t_{max}} \tag{11}$$

On the other hand, using the Mean Value Theorem and inequalities (5), (7), it is not difficult to show that $\sqrt{V(x)}$ is Lipschitz continuous in x and satisfies

$$\left| \sqrt{V(x^1)} - \sqrt{V(x^2)} \right| \quad \leq \quad \frac{c_4}{\sqrt{c_1}} \|x^1 - x^2\|$$

for all x^1, $x^2 \in \mathbb{R}^n$. Therefore,

$$\sqrt{V(x(t))} \quad \leq \quad \frac{c_4}{\sqrt{c_1}} \|x(t) - r_i(t)\| + \sqrt{V(r_i(t))} \tag{12}$$

Evaluating (12) at $t = t_{i+1}$ and applying (8), (11) lead to

$$\begin{aligned} \sqrt{V(x(t_{i+1}))} &\leq \frac{c_4}{\sqrt{c_1}} \|x(t_{i+1}) - r_i(t_{i+1})\| + \sqrt{V(r_i(t_{i+1}))} \\ &\leq \frac{c_4}{\sqrt{c_1}} \|x(t_{i+1}) - r_i(t_{i+1})\| + e^{-\frac{c_3}{2c_2}\Delta t_i} \sqrt{V(r_i(t_i))} \\ &\leq \epsilon K + e^{-\frac{c_3}{2c_2}\Delta t_i} \sqrt{V(r_i(t_i))} \end{aligned}$$

where $K = \frac{2c_4 L_2}{\sqrt{c_1}} \Delta t_{max} e^{L_1 \Delta t_{max}}$. Since $r_i(t)$ and $x(t)$ are starting from the same initial condition, i.e., $r_i(t_i) = x(t_i)$, we have

$$\sqrt{V(x(t_{i+1}))} \leq \epsilon K + e^{-\frac{c_3}{2c_2}\Delta t_i} \sqrt{V(x(t_i))} \tag{13}$$

Applying (13) recursively to the time intervals $[t_{k-1}, t_k]$, $k = i, i-1, \ldots, 1$, we have

$$\sqrt{V(x(t_{i+1}))} \leq \frac{\epsilon K}{1 - e^{-\frac{c_3}{2c_2}\Delta t_{min}}} + e^{-\frac{c_3}{2c_2} t_{i+1}} \sqrt{V(x(t_0))}$$

From the formula above and (5), it is apparent that $x(t_n)$ are bounded for all $n = 1, 2, \cdots$. Moreover,

$$\lim_{n\to\infty} \|x(t_n)\| \quad \leq \quad \frac{\epsilon K}{\sqrt{c_1} - \sqrt{c_1} e^{-\frac{c_3}{2c_2}\Delta t_{min}}}$$

■

Remark 3: Proposition 1 shows that for a sufficiently small computational delay, the closed-loop system is practically stable, and the stability error is proportional to the computational error. If the computation is perfect, i.e., $\epsilon = 0$,

then the system is asymptotically stable for any computational delay, in agreement with the Principle of Optimality.

Remark 4: If the computational time is not small enough to meet the requirement $\Delta t_{max} \leq \Delta T$, it is quite possible that the closed-loop system becomes unstable. From (10), it is easy to see that when Δt_i is too large, the state $x(t)$ will be pushed away from the reference $r_i(t)$ and the distance turns to infinity as time advances. From the derivation, the value of ΔT only depends on the system parameters (L_1, L_2, L_3). It means that, for a given nonlinear control system (Problem B), there is an *inherent* time constant. Any computational delay less than this time constant will not destroy stability.

Example 2: Consider the following LQR problem

$$\begin{cases} \text{Minimize} & J[x(\cdot),u(\cdot)] = & \int_0^\infty x_1^2(t) + 0.5x_2^2(t) \\ & & +0.25u^2(t)\ dt \\ \text{Subject to} & \dot{x}_1(t) = & x_2 \\ & \dot{x}_2(t) = & 2x_1 - x_2 + u \\ & (x_1(0), x_2(0)) = & (-1, 4) \end{cases}$$

A feedback controller based on Algorithm I with $N = 10$ indicates that the solution compares well with a Riccati-based solution (i.e. errors are within 10^{-4}). The average run time for our algorithm is 0.075 seconds. A simulation result is shown in the left plot in Fig. 3; this corresponds to a deliberate introduction of a computational delay of $\Delta t_i = 0.4$ seconds. Clearly, the system continues to be stable indicating that our algorithm works even if the computation is sluggish. As illustrated in the right plot in Fig. 3, if $\Delta t_i = 0.46$ seconds, the system becomes unstable. This demonstrates the result of Proposition 1, which indicates the existence of a stability threshold for computational delay.

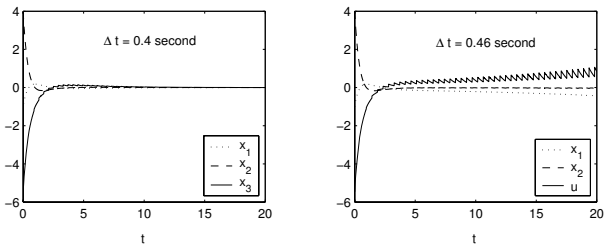

Fig. 3. Closed-loop performance with different computational delays.

C. Fixed Sampling Frequency

In Algorithm I we implicitly assumed that the measurement can be taken at any time. In some applications, the sampling frequency is fixed and hence, in this section, we modify the previous design to cope with this situation.

Let Δt be the fixed sampling period and $t_i = t_{i-1} + \Delta t$ be the sampling points. In Algorithm I, over any sampling interval, $[t_i, t_{i+1}]$, the optimal control is calculated from the previous run with $x(t_{i-1})$ as the initial condition. This introduces a computational delay. In the case of a fixed sampling frequency, we use a prediction of the state to alleviate the effects of computational delay as follows: At $t = t_0$ we apply the optimal open-loop controller to the system for the first sampling period $[t_0, t_1]$. At the same time, we propagate the trajectory under the action of this control. Thus, the state at $t = t_1$ is predicted. Denote the predicted state as $\hat{x}_1$. In the ideal case (i.e. without disturbance and propagation error), $x(t_1) = \hat{x}_1$. Let t_p be the computation time for the propagation. Normally t_p is very small; therefore, at time $t_0 + t_p \ll t_1$, we already have a prediction $\hat{x}_1$. Thus, at $t = t_0 + t_p$, we begin the computation of the optimal controller, $u_1^*(t)$, with $\hat{x}_1$ as the initial condition. Let t_c be the computational time and $t_p + t_c$ be less that Δt. Then, at $t = t_1$, this control is applied to the system over the time interval $[t_1, t_2]$, and the entire process is repeated.

Assumption 5: During any sampling interval $[t_i, t_{i+1}]$, it is assumed that $t_p + t_c \leq \Delta t$. That is, the prediction and the computation can be done in advance of the next sampling time.

Under A5, we formulate the following algorithm.

Algorithm II (Fixed Sampling Frequency)

Initialization:

1. Choose the discretization parameter N and the first sampling time t_1.
2. Collect the initial state $x_0 = x(t_0)$ and set $i = 1$.
3. Use the PS method to calculate the optimal controller $u_0(x_0, t)$ and apply it to the system till $t_1 = t_0 + \Delta t$.

Main Algorithm:

1. Propagate the system dynamics with the control $u_{i-1}(t)$ with $x(t_{i-1})$ as the initial condition to time $t = t_i$. Let $\hat{x}(t_i)$ be the predicted state at time t_i.
2. Use the PS method to calculate the optimal controller $u_i(\hat{x}(t_i), t)$ with $\hat{x}(t_i)$ as the initial condition.
3. Set $t_{i+1} = t_i + \Delta t$. During time period $[t_i, t_{i+1}]$, apply the controller $u_i(\hat{x}(t_i), t)$.
4. Collect the new measurement $x(t_i)$.
5. Set i = i+1 and go to step 1.

Assumption 6: There are positive constants δ, ϵ, such that for all i

$$\begin{aligned} \|x(t_i) - \hat{x}(t_i)\| &\leq \delta \\ \|u_i(\hat{x}(t_i), t) - u_i^*(\hat{x}(t_i), t)\| &\leq \epsilon \qquad \forall\ t \in [t_i, t_{i+1}] \end{aligned}$$

where $u_i^*(\hat{x}(t_i), t)$ is the optimal controller with $\hat{x}(t_i)$ as the initial condition.

Remark 5: As in A4, ϵ again represents the computational error in the calculation of the optimal controller while δ is used to represent the totality of propagation errors, measurement noise and disturbances that cause a deviation between the predicted state and the real state.

By a proof similar to that of Proposition 1, the following stability result can be obtained.

Proposition 2: Let the controller be designed according to Algorithm II. Under Assumption 1-2-3 and A5-6, for any initial condition $x(t_0)$, the state of the closed-loop system is bounded for all $t \in [t_0, \infty)$. Moreover,

$$\lim_{t\to\infty} \|x(t)\| \leq (\epsilon + L_3\delta)B(\Delta t)$$

where $B(\Delta t) = \frac{c_4 L_2 \Delta t e^{L_1 \Delta t}}{\sqrt{c_1}\left(1 - e^{-\frac{c_3}{2c_2}\Delta t}\right)}$

Remark 6: A major difference between Algorithm I and II is that, the stability of Algorithm II does not depend on the time constant of the system. The prediction step eliminates the computational delay in Algorithm I, and ensures stability even when the computation time is longer than the time constant of the system. In terms of accuracy or stability error, Algorithm I normally outperforms Algorithm II, since, given the same computational error ϵ, Algorithm I provides a smaller sampling period than Algorithm II.

IV. Nonlinear Stabilization of an Inverted Pendulum

We take the nonlinear model of an inverted pendulum [3]

$$\begin{aligned} \dot{x}_1 &= x_2,\ \dot{x}_2 = \frac{\psi_1(x,u)}{\psi_2(x)},\ \dot{x}_3 = x_4 \\ \dot{x}_4 &= \frac{3}{2L}\left(\frac{\cos(x_3)}{\psi_2(x)}\psi_1(x,u) + g\sin(x_3)\right) \end{aligned}$$

where $\psi_2(x) = M + m - \frac{3}{4}m\cos(x_3)^2$, $\psi_1(x,u) = -bx_2 - \frac{1}{2}mLx_4^2\sin(x_3) + \frac{3}{4}mg\cos(x_3)\sin(x_3) + u$ and:

M	mass of the cart	0.5kg
m	mass of the pendulum	0.5kg
b	friction on the cart	0.1N/m/sec
L	length of the pendulum	0.6m
g	gravitational acceleration	9.8/m/sec/sec
u	applied force (control)	[-10 10]N

The cost function to be minimized is

$$J[x(\cdot),u(\cdot)] = \int_0^\infty \sum_{i=1}^4 x_i^2(t) + u^2(t)\ dt$$

For Algorithm I, with $N = 15$, the closed-loop system is stable over 100 runs with randomly chosen initial conditions within an arbitrarily chosen bound of $-0.2 \le x_i(0) \le 0.2$, $i = 1,\ldots,4$. The average stability error over 100 runs, i.e., $\sum_{k=1}^{100} \lim_{t\to\infty} \|x^k(t)\|/100$, is within 10^{-5}, and the average run time for each iteration is about 0.08 seconds. A typical state-control history is plotted in Fig. 4 along with the open-loop optimal control for the same initial condition. Since an analytic solution is unavailable, we compare the closed-loop solution with an open-loop PS solution corresponding to 100 nodes. It is clear from Figure 4 that the difference between the closed-loop and the open-loop optimal control is very small. A simulation result for Algorithm II, with a sampling

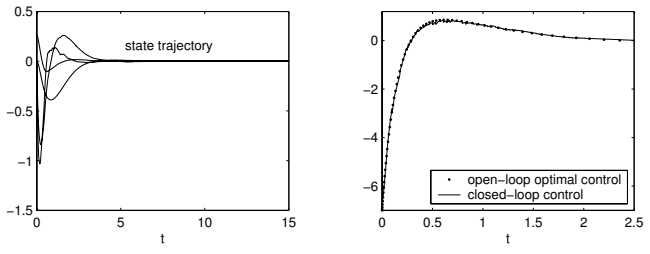

Fig. 4. Performance of a inverted pendulum under a controller constructed by Algorithm I

period of $\Delta t = 0.2$ seconds and $N = 15$, is shown in Figure 5. An impulse disturbance which instantly changes the state of the system is added at time $t = 11.6$. The simulation shows that the closed-loop controller automatically rejects this disturbance.

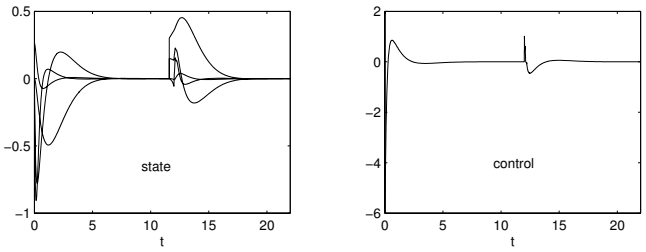

Fig. 5. Performance of a inverted pendulum under a controller constructed by Algorithm II. A disturbance is added at $t = 11.6$.

References

[1] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, second edition, Dover, 2001.

[2] G. Elnagar, M. A. Kazemi and M. Razzaghi, The pseudospectral Legendre method for discretizing optimal control problems, *IEEE Trans. Automat. Contr.* Vol. 40, pp. 1793-1796, 1995.

[3] F. Fahroo and I. M. Ross, Radau pseudospectral methods for infinite-horizon nonlinear optimal control problems, *Proc. AIAA Guid., Nav. and Control Conf.*, San Franscro, CA, 2005.

[4] A. Fleming, Real-time optimal slew maneuver design and control, Astronautical Engineer's Thesis, US Naval Postgraduate School, December 2004.

[5] Q. Gong, W. Kang and I. M Ross, A pseudospectral method for the optimal control of constrained feedback linearizable systems, *to appear in IEEE Trans. Automat. Contr.*, 2006.

[6] Q. Gong and I. M. Ross, Autonomous pseudospectral knotting methods for space mission optimization, *AAS/AIAA Space Flight Mechanics Conference,* Tampa, FL, 2006.

[7] A. M. Hawkins, T. R. Fill, R. J. Proulx, E. M. Feron, Constrained Trajectory Optimization for Lunar Landing, *AAS Spaceflight Mechanics Meeting*, Tampa, FL, January 2006, AAS 06-153.

[8] A. Isidori, *Nonlinear Control Systems. Springer-Verlag,* 1995.

[9] H. Khalil, *Nonlinear systems*, Macmillan Publishing Company, New York, 1992.

[10] S. Kothare and M. Morari, Contractive model predictive control for constrained nonlinear systems, *IEEE Trans. on Automat. Contr.*, Vol. 45, No. 6, pp. 1053-1071, 2000.

[11] D. Q. Mayne, J. B. Rawlings, C. V. Rao and P. O. M. Scokaert, Constrained model predictive control: stability and optimality, *Automatica*, Vol. 36, pp. 789-814, 2000.

[12] I. M. Ross, User's manual for DIDO: A MATLAB application package for solving optimal control problems, Technical Report 04-01.0, Tomlab Optimization Inc, February 2004.

[13] I. M. Ross and F. Fahroo, Pseudospectral methods for optimal motion planning of differentially flat systems, *IEEE Trans. on Automat. Contr.*, Vol. 49, No. 8, pp. 1410-1413, August 2004.

[14] I. M. Ross and F. Fahroo, Issues in the real-time computation of optimal control, *Mathematical and Computer Modelling*, Vol. 40, Pergamon Publication (to appear).

[15] I. M. Ross and F. Fahroo, A unified framework for real-time optimal control, *Proc. IEEE Conf. on Decision and Contr.*, Maui, December, 2003, pp. 2210-2215.

[16] I. M. Ross and C. D. D'Souza, Hybrid Optimal Control Framework for Mission Planning, *J. of Guid., Contr. and Dyn.,* Vol. 28, No. 4, pp. 686-697, 2005.

[17] P. Sekhavat, A. Fleming and I. M. Ross, Time-optimal nonlinear feedback control for the NPSAT1 spacecraft, *Proc. of the 2005 IEEE/ASME International Conf. on Advanced Intel. Mechatronics,* AIM 2005, 2428, Monterey, CA, 2005.

[18] P. Williams, Application of Pseudospectral Methods for Receding Horizon Control, *J. of Guid., Contr. and Dyn.*, Vol.27, No.2., 2004, pp.310-314.

[19] H. Yan and K. T. Alfriend, Three-axis Magnetic Attitude Control Using Pseudospectral Control Law in Eccentric Orbits, *AAS Spaceflight Mechanics Meeting*, Tampa, FL, January 2006, AAS 06-103.